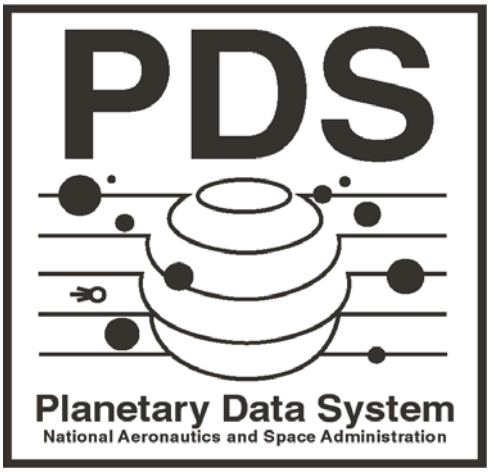
Planetary Data System

Validation Tool (VTool)
Test Plan

Alpha Test Phase II

September 12, 2006

Version 0.20060912



Jet Propulsion Laboratory
Pasadena, California

JPL D-xxxxx

CHANGE LOG

Revision	Date	Description	Author
Start Draft	2006-06-05	First Draft	R. Joyner
Draft Release 1	2006-07019	Requirements Version 1.1	R. Joyner
Draft Release 2	2006-09-12	Requirements Version 1.3	R. Joyner

TABLE OF CONTENTS

1. INTRODUCTION

This document is the Test Plan for the PDS Validation Tool (VTool) Version 0.3.0 -- Alpha Test Phase II.

1.1. Overview / Identification

VTool will read PDS labels and perform the following types of error checking:

- ODL Language checking – PDS labels are written using the PDS Object Description Language (ODL) (see Chapter 12 of [2]). ODL checking detects errors in the usage (syntax) of ODL, such as missing quotation marks, invalid characters in names, etc.
- Data Dictionary checking – The keywords and objects used within PDS labels follow certain PDS Standards [2]; as well as, rules specified by the design of the PDS data dictionary [3]. Data Dictionary checking determines when these rules (semantics) have been broken (e.g., specifying a COLUMN object within an IMAGE).

VTool has a simple command line interface which will operate on any of the PDS supported platforms / operating systems. VTool can verify one label at a time or a large number of labels. Messages resulting from the validation are written to a report file, to which the user can then refer to ascertain if the labels are PDS compliant.

1.2. Document Purpose

The purpose of this document is to document the testing approach used to formally verify that VTool functions and performs in accordance with the Level 5 requirements set forth in the Validation Tool (VTool) Requirements Document [1].

1.3. Document Scope

This document is intended to cover all aspects of testing Version 1.0 of the VTool application. The scope of testing will cover the capabilities which exist and which are defined by the VTool Requirements Document [1].

1.4. Applicable Documents

[1] Validation Tool Requirements, July 25, 2006, Version 1.3.

[2] Planetary Data System Standards Reference, February 1, 2006, Version 3.7, JPL D-7669, Part 2. (http://pds.jpl.nasa.gov/documents/sr/)

[3] Planetary Science Data Dictionary Document, August 28, 2002, Planetary Data System, JPL D-7116, Rev. E. (http://pds.jpl.nasa.gov/documents/psdd/psdd.pdf)

[4] Validation Tool (VTool) Procedure (JPL-D xxxx), September 12, 2006, Version 0.20060912.

[5] Validation Tool (VTool) Test Report Alpha Test Phase II, September 13, 2006.

2. TEST MANAGEMENT

2.1. Test Approach / Phases

This section lists the approach used to test at the various phases of the development life cycle. There will be three phases of testing for the VTool application or its components:

1. Development Testing
    a. Unit Testing – analytical studies and tests performed at the code level.
    b. Integration Testing – tests performed at the application level.
2. Beta Testing – a phased set of testing involving end-user participation.
3. Acceptance Testing – final integration testing involving end-user acceptance.

The Beta and Acceptance testing phases will represent two formal review points where success will be indicative of having achieved a quality product.

When a software test failure occurs during the execution of the various test levels, the general policy is that the particular test will be repeated, after correction of errors. If necessary, depending on the error types and conditions, regression testing may be performed for the change.

2.1.1. Development Testing – Unit Tests

Development Testing will consist of two principle types of tests; Unit Tests and Integration Tests.

2.1.1.1. Unit Tests

The objective of Unit testing is to isolate each part of the application and show that the individual components function correctly. Unit Tests will carried out by the EN development team and will be:

1. Developed to exercise the interface and functionality of a single class.
2. Exercised by the developers at build time
3. Allow developers to spot immediate detection of coding anomalies.
4. Included with the source code providing a good source of documentation and enabling on-site testing.
5. Built and managed with Junit (testing framework).

The set of tests performed during Unit Testing are documented within the source code of each class. Unit Testing represents the majority of the testing that will be performed against the VTool application and its components.

2.1.1.2. Integration Tests

The objective of Integration Testing is to catch a class of errors which cannot be found by Unit Testing (i.e., errors which relate to the interaction / aggregation of different program components / classes). Integration Testing assures that aggregates of units perform accurately together (i.e., verifies the proper integration / aggregation of the components / classes to realize major functions of the VTool application).

Integration Testing starts after the successful completion of Unit Testing. Integration Tests will carried out by the EN Operations team (Ops Team) and will be:

1. Limited to testing the application (e.g., the integration of classes into an application)
2. Exercised by the EN Ops team (e.g., a team not directly involved in development) using test cases developed by the Ops team.

The test cases used in Integration Testing will be described in Section x.0 of the VTool Software Test Procedure document [4]. Where feasible, the test suite will be automated and will consist of cross-platform tests against PDS-supported platforms. Additionally, installer package tests will be performed to assure proper installation of the application on all supported platforms.

2.1.2. Beta Testing

The objective of Beta Testing is to involve the end-user in formally reviewing and testing VTool. Following successful completion of Integration Testing, a Beta version of VTool along with test procedures will be made available to Node personnel for "local" testing.

Beta Testing will consist of a phased and iterative approach, initially including three Nodes (ATMOS, GEO and PPI) with increased Node participation in subsequent phases. Test cases, developed by the end-users / Nodes, which are indicative of unique testing strategies will be adopted into the suite of regression tests.

Beta Testing will continue until VTool can successfully operate against all of the tests in the regression test suite. Successful completion of Beta Testing will indicate that VTool is ready for Acceptance Testing.

The test cases used in Beta Testing will be incorporated into the regression test suite.

The EN will provide a mechanism for tracking bug / error reports from the end-users / Nodes.

2.1.3. Acceptance Testing

Acceptance Testing proves and validates the overall VTool functionality and performance in the operational environment. Acceptance Testing is carried out by the EN Operations Team. Should a failure occur during the execution of any of the test cases, the policy is that VTool will be corrected and the entire suite of regression tests will be repeated. Acceptance Testing will continue until VTool is able to successfully pass the entire suite of regression tests, as documented in the Software Test Procedure document [4]. Successful completion of Acceptance Testing will certify VTool as production ready (i.e., Version 1.0 of VTool will be released to the PDS community as part of the PDS Tool Suite).

Note: Following successful completion of Acceptance Testing, test cases, developed by the end-users / Nodes that are indicative of unique testing strategies, will be accepted for inclusion into the suite of regression tests to be run against the next release of VTool.

2.2. Test Cases

Test Cases will be developed for the Integration Tests and will become the baseline set of regression tests.

The Regression Test Suite will be comprised of a slew of individual test cases each designed to demonstrate a particular aspect of VTools ability to accurately validate a PDS label.

There are two possible outcomes in running VTool against a test case:

- **PASS** – In the case of a PDS compliant label, VTool correctly identifies the label as being compliant (i.e., no ODL anomalies are erroneous identified). In the case of a PDS non-compliant label, VTool correctly identifies those portions of the label that are non-compliant and accurately reports the anomalies).

- **FAIL** – In the case of a PDS compliant label, VTool incorrectly identifies the label as being non-compliant (i.e., ODL anomalies are identified and reported). In the case of a PDS non-compliant label, VTool was not able to identify the non-compliant portions and fails to report the anomalies).

All of the test cases in the regression suite will be written such that the expected result is to PASS validation given that VTool has the functionality required to validate the test case. In the case where VTool fails to accurately validate a test case, a LIEN will be identified against the test case.

Additional Test Cases will almost certainly be found during Beta Testing that will be incorporated into the full set of regression tests.

2.3. Test Monitoring / Reporting

2.3.1. Monitoring / Status

Execution of the test plan will be monitored to recognize deviations from the plan. Monitored test metrics will include percent of system tested, test cases executed/passed/failed/not executed, number and severity of problems reported, number of problems closed/resolved/retested.

2.3.2. Reporting / Status

A Test Report will be generated to document the test execution and test results. The Test Report will:

- Identify the items tested.
- Summarize the evaluation of the planned test items (expected versus actual, including the impact of variances).
- Indicate the version/revision level of the software tested.
- Indicate the environment in which testing took place.
- Reference the test scenario, test procedure, and test report, if they exist.
- Specify metrics that were monitored during the testing effort, if any.
- Contain a comprehensive test evaluation summary, including conclusions regarding the quality and stability of the product.

2.4. Test Timeline

This section identifies the major milestones that comprise VTool testing.

TBD

APPENDIX A SOFTWARE REQUIREMENTS TRACE MATRIX

This section provides a high level description of the Test Cases that will be used to validate the functionality defined by the Level 5 requirements specified in the VTool Requirements Document [1].

In the following matrix, an expected result of PASS indicates that VTool should accurately identify compliant labels as being compliant and non-compliant labels as being non-compliant. A LIEN is noted as the expected result for those instances where a test case identifies functionality not expected to be present in the version of VTool to be tested.

VTool
Requirements Trace Matrix

RD: L5.VAL.FR.21

The Tool shall be able to validate one or more PDS data products as the result of a single tool execution.

Test Case	Test Approach	Expected Result
VTT_EN_21-1	Validation of single PDS-3 compliant label	PASS

RD: L5.VAL.FR.29

The Tool shall be able to validate all PDS data products in a directory.

Test Case	Test Approach	Expected Result
VTT_EN_29-1	Validation of single PDS-3 compliant label having pairings of data product POINTERs and OBJECTs: a. ARRAY -- DATA/COMP_ARRAY-1.LBL b. DOCUMENT -- DOCUMENT/EDRSIS.LBL c. HEADER -- DATA/COMP_IMG-1.LBL d. HISTOGRAM -- DATA/COMBINED_DETACHED.LBL e. HISTORY -- DATA/COMP_HISTORY-1.LBL f. INDEX_TABLE -- INDEX/INDEX.LBL g. IMAGE -- DATA/COMP_IMG-1.LBL h. PALETTE -- DATA/COMP_PALETTE-1.LBL i. QUBE j. SERIES -- DATA/COMBINED_DETACHED.LBL k. SPECTRUM -- DATA/COMP_SPECTRUM-1.LBL l. SPICE_KERNEL -- DATA/COMBINED_DETACHED.LBL m. SPREADSHEET -- DATA/COMP_SPREADSHEET-1.LBL n. TABLE -- DATA/COMP_IMG-1.LBL L	PASS

RD: L5.VAL.FR.30

The Tool shall be able to traverse a directory tree and recursively validate the content of all directories.

Test Case	Test Approach	Expected Result
VTT_EN_30-1	Validation of multiple PDS-3 compliant labels (e.g., entire volume of labels): - attached, detached, combined detached labels	LIEN

	- AAREADME.TXT, ERRATA.TXT, VOLDESC.CAT, - catalog, index, data, document, and label directories	

RD: L5.VAL.FR.22

The Tool shall be able to validate a PDS data product that has been constructed with one of the following methods.
a) Attached Label
b) Detached Label
c) Combined Detached Label

A definition and examples of each of the above methods can be found in chapters 4 and 5 of the *PDS Standards Reference* [2].

Test Case	Test Approach	Expected Result
VTT_EN_22-1	Validation of single PDS-3 compliant attached label	PASS
VTT_EN_22-2	Validation of single PDS-3 compliant detached label	PASS
VTT_EN_22-3	Validation of single PDS-3 compliant combined detached label	PASS

RD: L5.VAL.FR.23

The Tool shall be able to merge the contents of label fragments referenced by ^STRUCTURE pointers with the contents of the parent label when validating a PDS label.

Test Case	Test Approach	Expected Result
VTT_EN_23-1	Validation of single PDS-3 compliant label having single STRUCTURE pointer	PASS
VTT_EN_23-2	Validation of single PDS-3 compliant label having multiple STRUCTURE pointers	PASS
VTT_EN_23-3	Validation of single PDS-3 compliant label having multiple (cascading) STRUCTURE pointers	PASS
VTT_EN_23-4	Validation of single PDS-3 non-compliant label having multiple STRUCTURE pointers where the STRUCTURE pointers span OBJECT / END_OBJECT definitions	PASS

RD: L5.VAL.FR.27

The Tool shall validate a PDS label fragment as it would a PDS label with the following exceptions:

a) An SFDU label must not be contained in the label fragment.
b) A PDS_VERSION_ID statement must not be contained in the label fragment.
c) File characteristic elements must not be contained in the label fragment.
d) An END statement must not be contained in the label fragment.

Test Case	Test Approach	Expected Result
VTT_EN_27-1	Validation of single PDS-3 compliant label having multiple (cascading) STRUCTURE pointers	PASS
VTT_EN_27-2	Validation of single PDS-3 compliant label which references a label fragment having an SFDU	PASS
VTT_EN_27-3	Validation of single PDS-3 compliant label which references a label fragment having a PDS_VERSION_ID statement	PASS
VTT_EN_27-4	Validation of single PDS-3 compliant label which references a label fragment having file characteristic elements	PASS
VTT_EN_27-5	Validation of single PDS-3 compliant label which references a label fragment having an END statement	PASS

RD: L5.VAL.FR.28

The Tool shall identify files having an extension of FMT as a PDS label fragment.

Test Case	Test Approach	Expected Result
VTT_EN_28-1	Validation of single PDS-3 compliant label which references a label fragment having an extension of FMT	PASS

RD: L5.VAL.FR.32

The Tool shall accept the following as input for specifying the data product(s) to be validated:

a) File Specification(s)
b) Directory Specification(s)
c) Uniform Resource Locator(s) (URL)

Test Case	Test Approach	Expected Result
VTT_EN_32-1	Validation of single PDS-3 compliant label as specified by the file specification	PASS
VTT_EN_32-2	Validation of single PDS-3 compliant label as specified by a directory specification	LIEN
VTT_EN_32-3	Validation of single PDS-3 compliant label as specified by a Uniform Resource Locator(s) (URL) specification	LIEN

RD: L5.VAL.FR.33

The Tool shall accept the following as input for specifying the instance(s) of the PSDD to be used for validation:

a) File Specification(s)
b) Uniform Resource Locator(s) (URL)

Test Case	Test Approach	Expected Result
VTT_EN_33-1	Validation of single PDS-3 compliant label using multiple instances of the PSDD where each is specified by a file specification	PASS
VTT_EN_33-2	Validation of single PDS-3 compliant label using multiple instances of the PSDD where each is specified by a Uniform Resource Locator(s) (URL) specification	LIEN

RD: L5.VAL.FR.1

The Tool shall syntactically validate a PDS label as specified in chapter 12 of the *PDS Standards Reference* [2].

As noted in chapter 12 of the Standards Reference, syntactic validation of PDS labels is limited to the PDS specific implementation of the ODL specification.

Test Case	Test Approach	Expected Result
VTT_EN_1-1	Validation of single PDS-3 compliant full label having: (1) SFDU_HEADER (2) OBJECTs (3) GROUPs (4) ELEMENTs (5) SYMBOL string (6) COMMENTs (7) series / set notation (8) integer based notation (9) real number notation (10) date / time notation (11) day of year / time notation (12) units designation (13) format effectors	PASS

	(14) file characteristics (15) use of hyphen in standard values	
VTT_EN_1-2	Validation of single PDS-3 compliant partial label having: (1) OBJECTs (2) ELEMENTs (3) SYMBOL string (4) COMMENTs (5) series / set notation (6) integer based notation (7) real number notation (8) date / time notation (9) day of year / time notation (10) units designation	PASS
VTT_EN_1-3	Validation of single PDS-3 non-compliant full label having: (1) missing equal sign (2) unquoted ELEMENT containing special characters (3) unmatched double quotes (4) unmatched OBJECT definition (5) unmatched GROUP definition (6) unmatched SYMBOL string (7) unmatched series / set notation (e.g., {1,2,3} and (1,2,3)) (8) unmatched series / set notation (advanced) (9) malformed integer based notation (e.g., 2#1001011#) (10) malformed real number notation (e.g., 1.234E2) (11) malformed date / time notation (e.g., 2006-12-31T23:59:59) (12) malformed day of year / time notation (e.g., 2006-365T23:59:59) (13) malformed binary/octal/hex notation (14) malformed units designation (e.g., <seconds>) (15) malformed comments (16) malformed keywords / keyword_values (17) missing / malformed line terminators (18) format effectors (19) malformed SFDU (20) record formats / file characteristics (21) duplicate keywords (22) missing END statement	PASS
VTT_EN_1-4	Validation of single PDS-3 non-compliant full label having: (1) mismatched POINTER to OBJECT definition (2) an OBJECT definition is not referenced by a POINTER (3) a GROUP definition is referenced by a POINTER (4) a POINTER references a non-existent OBJECT (5) a GROUP definition contains a data location pointer	PASS
VTT_EN_1-5	Validation of single PDS-3 non-compliant label; where: (1) a GROUP definition exists within a parent OBJECT definition (2) a GROUP definition contains a child OBJECT definition (3) a GROUP definition contains a child GROUP definition (4) a GROUP definition is not defined in the PSDD (5) ELEMENTs within a specific GROUP definition are not defined in the PSDD (6) a GROUP definition is specified more than once	PASS

RD: L5.VAL.FR.24

The Tool shall verify that the characters in a PDS label belong to a limited subset of the standard 7-bit ASCII character set as follows:

a) All characters in the range of 32 through 127 (decimal).
b) The line feed character (10 decimal).
c) The carriage return character (13 decimal).

The remaining 7-bit ASCII characters (1-9, 11, 12, and 14-31 decimal, which includes the horizontal and vertical tab and form feed

characters) are not permitted in PDS labels.

Test Case	Test Approach	Expected Result
VTT_EN_24-1	Validation of single PDS-3 compliant label having a DESCRIPTION containing: (1) all characters in the range of 32 through 127 (decimal) (2) the line feed character (10 decimal) (3) the carriage return character (13 decimal)	PASS
VTT_EN_24-2	Validation of single PDS-3 non-compliant label having a DESCRIPTION containing: (1) all characters in the range of 32 through 127 (decimal) (2) the line feed character (10 decimal) (3) the carriage return character (13 decimal) (4) illegal characters (1 through 9 decimal) (5) illegal characters (11 and 12 decimal) (6) illegal characters (13 through 31 decimal)	PASS
VTT_EN_24-3	Validation of single PDS-3 non-compliant label having a DESCRIPTION containing: (1) all characters in the range of 128 through 255 (decimal)	PASS

RD: L5.VAL.FR.25

The Tool shall verify that all lines in a PDS label are terminated with a carriage return character followed by a line feed character.

Test Case	Test Approach	Expected Result
VTT_EN_25-1	Validation of single PDS-3 compliant label having all lines terminated with <CR><LF>	PASS
VTT_EN_25-2	Validation of single PDS-3 non-compliant label having multiple lines terminated with: (1) only the line feed character (10 decimal) (2) only the carriage return character (13 decimal)	PASS

RD: L5.VAL.FR.34

The Tool shall verify that a date/time value in a PDS label is valid as specified in chapter 7 of the PDS Standards Reference [2].

Test Case	Test Approach	Expected Result
VTT_EN_34-1	Validation of single PDS-3 non-compliant full label having: (1) malformed date / time notation (e.g., 2006-12-31T23:59:59) (2) malformed day of year / time notation (e.g., 2006-365T23:59:59)	PASS

RD: L5.VAL.FR.2

The Tool shall semantically validate a PDS label according to the constructs as specified in one or more PDS compliant data dictionaries.

Test Case	Test Approach	Expected Result
	This requirement will be satisfied by the successful demonstration of the child requirements	

RD: L5.VAL.FR.2.1

The Tool shall verify that all objects in the PDS label exist as specified in one or more PDS compliant data dictionaries.

Test Case	Test Approach	Expected Result
VTT_EN_2_1-1	Validation of single PDS-3 compliant label having multiple OBJECTs that are defined across multiple instances of the PSDD: (1) locally defined object having required & optional elements	PASS

	(2) merge mutually exclusive OBJECT & ELEMENT definitions from two dictionaries	
VTT_EN_2_1-2	Validation of single PDS-3 compliant label having multiple OBJECTs that are defined across multiple instances of the PSDD: (1) locally defined object having required & optional elements (2) merge mutually exclusive OBJECT & ELEMENT definitions from two dictionaries	PASS
VTT_EN_2_1-3	Validation of single PDS-3 compliant label having multiple OBJECTs that are defined across multiple instances of the PSDD: (1) locally defined object having required & optional elements (2) overwrite locally defined object and element definitions from two dictionaries	PASS
VTT_EN_2_1-4	Validation of single PDS-3 compliant label having multiple OBJECTs that are defined across multiple instances of the PSDD: (1) locally defined object having required & optional elements (2) merge locally defined object and element definitions from three dictionaries	PASS
VTT_EN_2_1-5	Validation of single PDS-3 compliant label having multiple OBJECTs that are defined across multiple instances of the PSDD: (1) locally defined object having required & optional elements (2) merge locally defined object and element definitions from two dictionaries; overwrite object & element definitions from the merge dictionaries into primary dictionary.	PASS

Note: This requirement is under scrutiny as OBJECT definitions should only exist in the psdd.

RD: L5.VAL.FR.2.2

The Tool shall verify that all elements in the PDS label exist as specified in one or more PDS compliant data dictionaries.

Test Case	Test Approach	Expected Result
VTT_EN_2_2-1	Validation of single PDS-3 compliant label having multiple ELEMENTs ; at least one of which is defined in an instance of the PSDD.	PASS
VTT_EN_2_2-2	Validation of single PDS-3 compliant label having multiple ELEMENTs that are defined across multiple instances of the PSDD.	PASS
VTT_EN_2_2-3	Validation of single PDS-3 non-compliant label having multiple ELEMENTs; at least one of which is not defined in an instance of the PSDD: (3) non-defined element having standard_value of data_type = integer (4) non-defined element having standard_value of data_type = <not> integer	PASS
VTT_EN_2_2-4	Validation of single PDS-3 non-compliant label having multiple ELEMENTs; at least one of which is: (1) not defined as either Required or Optional for the ELEMENT (as defined in the PSDD) (2) missing when is defined as Required for the ELEMENT (as defined in the PSDD) (3) missing when is defined as Optional for the ELEMENT (as defined in the PSDD).	PASS
VTT_EN_2_2-5	Validation of single PDS-3 compliant label having multiple ELEMENTs that are defined across multiple instances of the PSDD: (3) locally defined elements (4) overwrite locally defined element definitions from two dictionaries	PASS

RD: L5.VAL.FR.2.3

The Tool shall verify that all element values are valid as specified as specified in one or more PDS compliant data dictionaries, including:
a) That all element values are consistent with the specified element value type.

b) That the length of all non-numeric element values is within the specified length limit.
c) That all element values constrained by enumerated lists (STATIC and DYNAMIC) are allowed values.
d) That all numeric element values are within the specified range.

Test Case	Test Approach	Expected Result
VTT_EN_2_3-1	Validation of single PDS-3 compliant label having multiple ELEMENTs that are defined across multiple instances of the PSDD.	PASS
VTT_EN_2_3-2	Validation of single PDS-3 non-compliant label having ELEMENTs that are defined across multiple instances of the PSDD; where: (1) the element value is not consistent with the element value type (as defined in the PSDD) (2) the length of the element value exceed the specified length limit (as defined in the PSDD) (3) the element value is not contained within the enumerated lists (as defined in the PSDD) (4) the element value is not within the specified range of values (as defined in the PSDD)	PASS
VTT_EN_2_3-3	Validation of single PDS-3 non-compliant label having ELEMENTs and OBJECTs that are defined across multiple instances of non-compliance: (1) missing keywords (2) aliased keywords (3) keyword_values too large (4) set / series notation (5) illegal use of comments (6) invalid use of FILE object (7) invalid keyword_values (8) data file extension (9) invalid use of symbol strings (10) missing keyword_values	PASS
VTT_EN_2_3-4	Validation of single PDS-3 compliant label having element values constrained by: (1) STATIC enumerated lists (2) DYNAMIC enumerated lists (3) SUGGESTED enumerated lists (4) DEFINITION enumerated lists (5) FORMATION enumerated lists (6) RANGE enumerated lists (7) TEXT enumerated lists	PASS

RD: L5.VAL.FR.2.4

The Tool shall verify that all required objects and elements as specified in the PSDD, exist in the PDS label.

Test Case	Test Approach	Expected Result
VTT_EN_2_4-1	Validation of single PDS-3 compliant label having multiple OBJECTs and ELEMENTs that are defined across multiple instances of the PSDD.	PASS
VTT_EN_2_4-2	Validation of single PDS-3 non-compliant label; where: (1) the child OBJECT definition is not consistent with the parent OBJECT definition (as defined in the PSDD)	PASS
VTT_EN_2_4-3	Validation of single PDS-3 non-compliant label; where: (1) the ELEMENT definition is not consistent with the parent OBJECT definition (as defined in the PSDD)	PASS

RD: L5.VAL.FR.3

The Tool shall verify that a data object exists when referenced by a pointer in the PDS label.

Test Case	Test Approach	Expected Result

VTT_EN_3-1	Validation of single PDS-3 compliant label having: (1) multiple pairs of POINTERs and OBJECTs referenced by POINTERs (2) a GROUP definition containing: - an "include" POINTER - a "related information" POINTER	PASS
VTT_EN_3-2	Validation of single PDS-3 compliant label; having: - mismatched POINTER to existing file location	PASS

RD: L5.VAL.FR.19

The Tool shall verify that a file exists when referenced by a pointer in the PDS label.

Test Case	Test Approach	Expected Result
VTT_EN_19-1	Validation of single PDS-3 compliant label having multiple pairings of POINTERs and OBJECTs referenced by POINTERs; and files referenced by POINTERs: (1) STRUCTURE pointer (2) DESCRIPTION pointer (3) NOTE pointer (e.g., USAGE_NOTE) (4) TEXT pointer (5) CATALOG pointers (e.g., DATA_SET_CATALOG, MISSION_CATALOG) (6) DATA OBJECT pointers a. ARRAY b. DOCUMENT (e.g., ASCII_DOCUMENT, TEX_DOCUMENT) c. GAZETTEER (e.g., GAZETTEER_TABLE) d. HEADER (e.g., IMAGE_HEADER) e. HISTOGRAM (e.g., IMAGE_HISTOGRAM) f. HISTORY g. INDEX_TABLE (e.g., MY_INDEX_TABLE) h. IMAGE (e.g., MY_IMAGE) i. PALETTE j. QUBE k. SERIES (e.g., TIME_SERIES) l. SPECTRUM (e.g., IMAGE_SPECTRUM) m. SPICE_KERNEL n. SPREADSHEET o. TABLE (e.g., INDEX_TABLE, GAZETTEER_TABLE, MY_TABLE)	PASS
VTT_EN_19-2	Validation of single PDS-3 non-compliant label; where: (1) file referenced by a STRUCTURE pointer is not co-located with label or in LABEL directory (2) file referenced by DESCRIPTION pointer is not co-located with label or in DOCUMENT directory (3) file referenced by NOTE pointer is not co-located with label or in DOCUMENT directory (4) file referenced by TEXT pointer is not co-located with label or in DOCUMENT directory (5) file referenced by CATALOG pointer is not co-located with label or in CATALOG directory (6) file referenced by INDEX_TABLE pointer is not co-located with label or in INDEX directory (7) file referenced by DOCUMENT pointer is not co-located with label or in DOCUMENT directory (8) the file referenced by the following DATA OBJECT pointers is not co-located with label: a. ARRAY b. HEADER c. HISTOGRAM d. HISTORY e. IMAGE f. PALETTE	LIEN

	g. QUBE h. SERIES i. SPECTRUM j. SPICE_KERNEL k. SPREADSHEET l. TABLE (only exception is INDEX_TABLE, MY_INDEX_TABLE)	
VTT_EN_19-3	Validation of single PDS-3 non-compliant label; where: (1) a POINTER references a data product file having too few records (2) a POINTER uses a non-standard location reference <RECORDS>	LIEN

RD: L5.VAL.FR.26

The Tool shall validate the following with respect to a Standard Formatted Data Unit (SFDU) present in a PDS label, as specified in chapter 16 of the PDS Standards Reference [2]: (L4.VAL.FR.1)

a) The length of each SFDU label is correct.
b) The location of each SFDU, within a PDS label is correct.
c) The Control Authority ID is valid for the given SFDU label class.
d) The Version ID is valid.
e) The Class ID is valid.
f) The Delimiter Type is valid.
g) The start marker in a K class SFDU label has an identical end marker.

Test Case	Test Approach	Expected Result
VTT_EN_26-1	Validation of single PDS-3 compliant label having a Version 3 SFDU_HEADER; where: (1) both SFDU and RECORD_TYPE indicate STREAM (2) both SFDU and RECORD_TYPE indicate FIXED_LENGTH (3) both SFDU and RECORD_TYPE indicate VARIABLE_LENGTH (not used) (4) both SFDU and RECORD_TYPE indicate UNDEFINED	LIEN
VTT_EN_26-2	Validation of single PDS-3 non-compliant label having: (1) contains a Version 2 SFDU_HEADER	LIEN
VTT_EN_26-3	Validation of single PDS-3 non-compliant label having: (1) a malformed Version 3 SFDU statement terminator (2) a missing SFDU statement terminator (3) a misplaced SFDU_HEADER	LIEN
VTT_EN_26-4	Validation of single PDS-3 non-compliant label having a Version 3 SFDU_HEADER; where: (1) the SFDU indicates STREAM, but the RECORD_TYPE = FIXED_LENGTH (2) the SFDU indicates FIXED_LENGTH, but the RECORD_TYPE = STREAM (3) the SFDU indicates UNDEFINED, but the RECORD_TYPE = STREAM (4) the SFDU indicates UNDEFINED, but the RECORD_TYPE = FIXED_LENGTH (5) the SFDU indicates VARIABLE_LENGTH, but the RECORD_TYPE = STREAM	LIEN
VTT_EN_26-5	Validation of single PDS-3 non-compliant label having a Version 3 SFDU_HEADER; where: (1) the length of the SFDU is incorrect (2) the control authority is specified incorrectly (3) the version id is specified incorrectly (4) the class id is specified incorrectly (5) the delimiter type is specified incorrectly (6) the start marker does not have an identical end marker (7) massively malformed SFDU	LIEN

RD: L5.VAL.FR.38

The Tool shall report the existence of a Standard Formatted Data Unit (SFDU), if encountered in a PDS label.

Test Case	Test Approach	Expected Result

VTT_EN_38-1	Validation of single PDS-3 compliant label: (1) having a Version 3 SFDU_HEADER (2) not having a Version 3 SFDU_HEADER	PASS

RD: L5.VAL.FR.35

The Tool shall verify that the PDS_VERSION_ID element is either:
a) The first line of a PDS label, if an SFDU is not present in the label.
b) The second line of a PDS label, if an SFDU is present in the label.

This requirement is derived from section 5.3.1 [2].

Test Case	Test Approach	Expected Result
VTT_EN_35-1	Validation of single PDS-3 compliant label: (1) having a Version 3 SFDU_HEADER (2) not having a Version 3 SFDU_HEADER	PASS
VTT_EN_35-2	Validation of single PDS-3 non-compliant label: (1) PDS_VERSION_ID as 1st line; SFDU as 2nd line (2) OBJECT = FILE as 1st line (3) STRUCTURE pointer as 1st line (which contains SFDU & PDS_VERSION_ID)	PASS

RD: L5.VAL.FR.36

The Tool shall verify that a PDS label contains the appropriate file characteristic elements as specified in section 5.3.2 of the *PDS Standards Reference* [2].

The file characteristic elements include RECORD_TYPE, RECORD_BYTES, FILE_RECORDS and LABEL_RECORDS.

Test Case	Test Approach	Expected Result
VTT_EN_36-1	Validation of single PDS-3 compliant label; where: (1) detached label; RECORD_TYPE = STREAM; (2) detached label; RECORD_TYPE = FIXED_LENGTH (3) detached label; RECORD_TYPE = VARIABLE_LENGTH (4) detached label; RECORD_TYPE = UNDEFINED (5) attached label; RECORD_TYPE = STREAM; (6) attached label; RECORD_TYPE = FIXED_LENGTH (7) attached label; RECORD_TYPE = VARIABLE_LENGTH (8) attached label; RECORD_TYPE = UNDEFINED	LIEN
VTT_EN_36-2	Validation of single PDS-3 non-compliant label; where: (1) RECORD_TYPE = STREAM; missing file characteristic elements (2) RECORD_TYPE = FIXED_LENGTH; missing file characteristic elements (3) RECORD_TYPE = VARIABLE_LENGTH; missing file characteristic elements (4) RECORD_TYPE = UNDEFINED; missing file characteristic elements	LIEN

RD: L5.VAL.FR.37

The Tool shall verify that a PDS label contains the appropriate file characteristic elements as specified in section 5.3.2 of the *PDS Standards Reference* [2].

The file characteristic elements include RECORD_TYPE, RECORD_BYTES, FILE_RECORDS and LABEL_RECORDS.

Test Case	Test Approach	Expected Result
VTT_EN_37-1	Note: This requirement is under scrutiny as the data product could itself be padded which would preclude VTool from accurately calculating record offsets.	LIEN